2017/18 CSC1023 Databases:

Project Assignment

**Module:** CSC1023 Databases

**Title:** Project Assignment

**Weight:**  30% of Module

**Set by:** Dr. David Cutting

**Moderated by:** Dr. Andrew McDowell

**Date Set:** 20th February 2018

**Submission Due:** 1700 (5.00pm) 16th April 2018

# Objectives

To demonstrate competency and understanding in:

* Creation of RDBMS databases using SQL from a provided schema
* Implementing constraints to ensure referential integrity
* Use of SQL for data manipulation and extraction
* Use of SQL for extraction of complex related data sets
* Connecting to and using an RDBMS from an external application
* Planning and executing the necessary changes to a program
* Considering and presenting future plans to fulfil business needs

# Deliverables

1. A number of SQL scripts to meet the requirements to (a) create, (b) populate, (c) validate, and (d) exploit a defined schema and data set
2. An updated Java program (source code) with highlighted changes made in accordance with the requirements and a report showing the changes and testing of this program
3. A report containing answers to questions pertaining to future expansion, modification, and protection of key database systems

Table of Contents

[Objectives 1](#_Toc510354261)

[Deliverables 1](#_Toc510354262)

[Introduction and Outline 3](#_Toc510354263)

[Scenario 3](#_Toc510354264)

[Assignment Components 3](#_Toc510354265)

[Individual Work 3](#_Toc510354266)

[A. Schema Implementation 4](#_Toc510354267)

[Submission Requirements for Part A (Schema Implementation) 4](#_Toc510354268)

[Entity Relationship Diagram 5](#_Toc510354269)

[Relationships (Cardinality) Description 6](#_Toc510354270)

[Data Dictionary 7](#_Toc510354271)

[Sample Data 20](#_Toc510354272)

[Exploiting the Database 21](#_Toc510354273)

[B. Java Program Update 22](#_Toc510354274)

[Task 22](#_Toc510354275)

[Deliverables for Part B (Java Program Update) 22](#_Toc510354276)

[C. Future Planning 23](#_Toc510354277)

[Task 23](#_Toc510354278)

[Deliverables for Part C (Future Planning) 23](#_Toc510354279)

[Reports 23](#_Toc510354280)

[Mark Allocation Scheme 24](#_Toc510354281)

[Final Submission 25](#_Toc510354282)

# Introduction and Outline

## Scenario

Wombat Widgets International (WWI) are a large international (over 5000 employees based in 16 countries, with manufacturing facilities in 3 countries) manufacturer of top quality widgets (small gadgets or mechanical devices).

You have been called into work with WWI on behalf of a large Professional Services firm who WWI have contracted to develop and consult on a range of database requirements they have.

## Assignment Components

1. Schema Implementation – 70%
2. Java Program Update – 10%
3. Future Planning – 20%

Read the instructions required for each section carefully especially note:

* What you are required to produce
* The format of this deliverable
* How it is to be organised and submitted

## Individual Work

This project assignment is an individual piece of work and so submitted work must be (a) new [never before submitted] work by the student alone and (b) not created in collusion with anyone else. Colluding with other students or any plagiarism including use of paid third-parties are academic offences under the University regulations.

# A. Schema Implementation

We have been tasked with implementing a new Human Resources (HR) and Payroll system for WWI. Following a detailed requirements gathering and analysis process a normalised schema has been generated. You must now implement this schema in the MySQL RDBMS.

Once implemented you must also populate the schema with some provided data and test it to ensure it meets the requirements. You will then also be required to write some queries to exploit the related entities within the database.

On submission a set of automated tests will be performed to validate the schema, and the ability to run some similar automated tests will be made available to you later in the process.

## Submission Requirements for Part A (Schema Implementation)

1. A file named schema.sql containing just the schema (table definitions)
2. A file names data.sql containing the INSERT statements to populate the example data
3. A file named query.sql containing the queries required one per line (10 queries in the one file, comments or whitespace are fine but nothing else – if you need to skip a query just have a *SELECT \* FROM Employee;* query or similar for the missed one – make sure to finish each query with a semi-colon ;)

## Entity Relationship Diagram

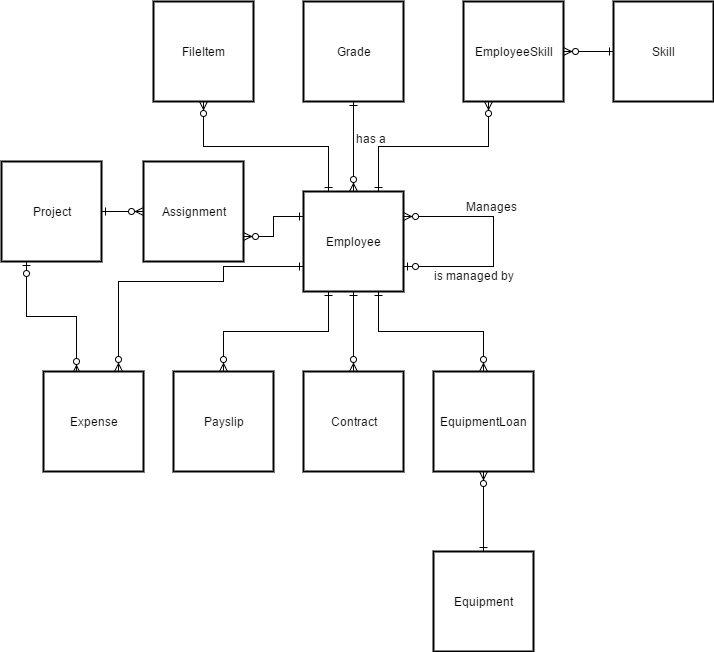


Figure 1. Entity Relationship Diagram (ERD)

## Relationships (Cardinality) Description

* An Employee may have zero or more FileItems (arbitrary items stored relating to the employee) and each FileItem must be linked to one Employee.
* An Employee must have a Grade of employment and each Grade of employment may have zero or more Employees.
* An Employee may have zero or more Skills and each Skill may be linked to zero or more Employees through the EmployeeSkill entity i.e. an Employee may have zero or more EmployeeSkills but each EmployeeSkill must be linked to one Employee, a Skill may have zero or more EmployeeSkills and each EmployeeSkill must be linked to one Skill.
* An Employee may have zero or more Assignments but each Assignment must have one Employee.
* A project may have zero or more Assignments but each Assignment must have one linked Project
* An Employee may have zero or more sub-ordinate Employees that they manage and an Employee may have zero or one line managers (another Employee who manages them).
* An Employee may have zero or more Expense claims, each Expense claim must be for one Employee. Expense claims may be linked against one project (or not) and so each Project may have zero or more Expense claims linked to it.
* An Employee may have zero or more Payslips and each Payslip must be linked to one Employee.
* An employee may have zero or more employment Contracts and each Contract must be linked to one Employee.
* An Employee may have zero or more EquipmentLoans and each EquipmentLoan must be linked to one Employee. Each item of Equipment may have zero or more EquipmentLoans recorded against it and each EquipmentLoan is for one piece of Equipment.

## Data Dictionary

There follows a data dictionary containing each of the tables that need created.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Database:** | Wombat HR | **Table Name:** | FileItem | **Key Field(s):** | ItemID |

**Related to:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name** | **Foreign Key** | **Table Name** | **Foreign Key** |
| Employee | EmployeeID |  |  |

**General description:**

|  |
| --- |
| A list of all the general/arbitrary items stored in relation to an employee (in the employee file) such as references, commendations, etc) which may be electronic files (stored on a share drive) or even physical items in a particular filing cabinet |

**Fields:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **R** | **I** | **U** | **Data Type** | **Length** | **Format Rule(s)** | **Default Value** | **Description** | **Example Data** |
| ItemID | Y | Y | Y | BIGINT |  |  | n/a | AUTO\_INCREMENT primary key | 1 |
| Title | Y | Y | N | VARCHAR | 255 |  | “” | Title of the item stored | “Reference from Previous Employer” |
| Location | Y | N | N | VARCHAR | 255 |  | “” | Location of the item (where stored) | “S drive in the employee folder” |
| EmployeeID | Y | Y | N | BIGINT |  |  |  | Employee ID FK on Employee.EmployeeID | 123 |

**Key:** R = Required (Y/N), I = Indexed (Y/N), U = Unique (Y/N)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Database:** | Wombat HR | **Table Name:** | Grade | **Key Field(s):** | GradeID |

**Related to:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name** | **Foreign Key** | **Table Name** | **Foreign Key** |
|  |  |  |  |

**General description:**

|  |
| --- |
| Collection of all the possible employment grades of Employees |

**Fields:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **R** | **I** | **U** | **Data Type** | **Length** | **Format Rule(s)** | **Default Value** | **Description** | **Example Data** |
| GradeID | Y | Y | Y | INT |  |  |  | AUTO\_INCREMENT ID for the grade, PRIMARY KEY for Grade | 1 |
| Title | Y | N | N | VARCHAR | 128 |  |  | Title for the grade | “Senior Consultant” |
| Code | Y | Y | Y | VARCHAR | 16 |  |  | Short code for the grade | “SC” |
| SpineMin | Y | N | N | INT |  |  | 1 | Minimum spine point | 10 |
| SpineMax | Y | N | N | INT |  |  | 1 | Maximum spine point | 16 |

**Key:** R = Required (Y/N), I = Indexed (Y/N), U = Unique (Y/N)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Database:** | Wombat HR | **Table Name:** | EmployeeSkill | **Key Field(s):** | EmployeeID  SkillID |

**Related to:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name** | **Foreign Key** | **Table Name** | **Foreign Key** |
| Employee | EmployeeID | Skill | SkillID |

**General description:**

|  |
| --- |
| Link table to link employees to skills |

**Fields:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **R** | **I** | **U** | **Data Type** | **Length** | **Format Rule(s)** | **Default Value** | **Description** | **Example Data** |
| EmployeeID | Y | Y | N | BIGINT |  |  |  | Employee ID FK on Employee.EmployeeID | 12345 |
| SkillID | Y | Y | N | BIGINT |  |  |  | Skill ID FK on Skill.SkillID | 45678 |
| DateAchieved | N | N | N | DATE |  |  |  | Date skill was achieved | 2018-01-01 |

**Key:** R = Required (Y/N), I = Indexed (Y/N), U = Unique (Y/N)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Database:** | Wombat HR | **Table Name:** | Skill | **Key Field(s):** | SkillID |

**Related to:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name** | **Foreign Key** | **Table Name** | **Foreign Key** |
|  |  |  |  |

**General description:**

|  |
| --- |
| Collection of possible skills for employees to have |

**Fields:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **R** | **I** | **U** | **Data Type** | **Length** | **Format Rule(s)** | **Default Value** | **Description** | **Example Data** |
| SkillID | Y | Y | Y | BIGINT |  |  |  | AUTO\_INCREMENT Skill ID, PRIMARY KEY | 45678 |
| Title | Y | Y | Y | VARCHAR | 255 |  |  | Title of Skill | “VBA Coding” |

**Key:** R = Required (Y/N), I = Indexed (Y/N), U = Unique (Y/N)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Database:** | Wombat HR | **Table Name:** | Project | **Key Field(s):** | ProjectID |

**Related to:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name** | **Foreign Key** | **Table Name** | **Foreign Key** |
|  |  |  |  |

**General description:**

|  |
| --- |
| Table of projects Wombat employees may be working on |

**Fields:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **R** | **I** | **U** | **Data Type** | **Length** | **Format Rule(s)** | **Default Value** | **Description** | **Example Data** |
| ProjectID | Y | Y | Y | BIGINT |  |  |  | AUTO\_INCREMENT Project ID, PRIMARY KEY | 1 |
| Title | Y | Y | N | VARCHAR | 255 |  |  | Project Title | “Build Super Widget” |
| Notes | N | Y | N | TEXT |  |  |  | Project Notes/Description | “Building a super widget to take over the world” |
| Internal | Y | N | N | BOOLEAN |  |  | 0 | Internal project flag | 1 |
| Sensitive | Y | N | N | BOOLEAN |  |  | 0 | Is a sensitive/confidential project | 1 |
| Started | Y | N | N | DATE |  |  |  | Date project started | 2017-06-01 |
| Ended | N | N | N | DATE |  |  |  | Date project ended (or NULL) | 2018-06-01 |

**Key:** R = Required (Y/N), I = Indexed (Y/N), U = Unique (Y/N)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Database:** | Wombat HR | **Table Name:** | Assignment | **Key Field(s):** | ProjectID  EmployeeID |

**Related to:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name** | **Foreign Key** | **Table Name** | **Foreign Key** |
| Project | ProjectID | Employee | EmployeeID |

**General description:**

|  |
| --- |
| Link table to show which employees are assigned to which projects |

**Fields:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **R** | **I** | **U** | **Data Type** | **Length** | **Format Rule(s)** | **Default Value** | **Description** | **Example Data** |
| ProjectID | Y | Y | N | BIGINT |  |  |  | Project ID FK on Project.ProjectID | 456 |
| EmployeeID | Y | Y | N | BIGINT |  |  |  | Employee ID FK on Employee.EmployeeID | 789 |

**Key:** R = Required (Y/N), I = Indexed (Y/N), U = Unique (Y/N)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Database:** | Wombat HR | **Table Name:** | Employee | **Key Field(s):** | EmployeeID |

**Related to:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name** | **Foreign Key** | **Table Name** | **Foreign Key** |
| Grade | GradeID | Employee | EmployeeID |

**General description:**

|  |
| --- |
| Employee table containing employee details |

**Fields:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **R** | **I** | **U** | **Data Type** | **Length** | **Format Rule(s)** | **Default Value** | **Description** | **Example Data** |
| EmployeeID | Y | Y | Y | BIGINT |  |  |  | AUTO\_INCREMENT Employee ID, PRIMARY KEY | 123 |
| Title | N | N | N | VARCHAR | 32 |  |  | Title of employee | “Rt. Hon. Professor” |
| FirstName | Y | Y | N | VARCHAR | 255 |  |  | First Name | “Bob” |
| LastName | Y | Y | N | VARCHAR | 255 |  |  | Last Name | “Smith” |
| Email | N | Y | Y | VARCHAR | 255 |  |  | Work email address | “bob@wombat.com” |
| Joined | N | N | N | DATE |  |  |  | Start Date | 2016-02-01 |
| Left | N | N | N | DATE |  |  |  | Finish Date (ex employees) | 2017-01-01 |
| Current | Y | Y | N | BOOLEAN |  |  | 1 | Current employee flag | 1 |
| Phone | Y | N | N | VARCHAR | 32 |  |  | Phone number | “02890974998” |
| GradeID | Y | N | N | INT |  |  |  | FK GradeID on Grade.GradeID | 6 |
| Manager | N | Y | N | BIGINT |  |  |  | FK Manager on Employee.EmployeeID | 123 |

**Key:** R = Required (Y/N), I = Indexed (Y/N), U = Unique (Y/N)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Database:** | Wombat HR | **Table Name:** | Expense | **Key Field(s):** | ExpenseID |

**Related to:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name** | **Foreign Key** | **Table Name** | **Foreign Key** |
| Employee | EmployeeID | Project | ProjectID |

**General description:**

|  |
| --- |
| Expenses raised by employees which may be linked to a project |

**Fields:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **R** | **I** | **U** | **Data Type** | **Length** | **Format Rule(s)** | **Default Value** | **Description** | **Example Data** |
| ExpenseID | Y | Y | Y | BIGINT |  |  |  | AUTO\_INCREMENT primary key | 663 |
| EmployeeID | Y | Y | N | BIGINT |  |  |  | FK EmployeeID on Employee.EmployeeID | 123 |
| ProjectID | N | Y | N | BIGINT |  |  |  | FK ProjectID on Project.ProjectID | 456 |
| Description | Y | Y | N | VARCHAR | 255 |  |  | Description of Expense | “Emergency garden gnomes” |
| Amount | Y | N | N | DOUBLE |  |  |  | Amount of expense | 59.99 |
| Paid | Y | N | N | BOOLEAN |  |  | 0 | Expense paid to employee flag | 1 |

**Key:** R = Required (Y/N), I = Indexed (Y/N), U = Unique (Y/N)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Database:** | Wombat HR | **Table Name:** | Payslip | **Key Field(s):** | PayslipID |

**Related to:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name** | **Foreign Key** | **Table Name** | **Foreign Key** |
| Employee | EmployeeID |  |  |

**General description:**

|  |
| --- |
| Record of an individual payment made to an employee (a pay slip) |

**Fields:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **R** | **I** | **U** | **Data Type** | **Length** | **Format Rule(s)** | **Default Value** | **Description** | **Example Data** |
| PayslipID | Y | Y | Y | BIGINT |  |  |  | AUTO\_INCREMENT primary key | 1234 |
| EmployeeID | Y | Y | N | BIGINT |  |  |  | FK EmployeeID on Employee.EmployeeID | 456 |
| Taxable | Y | N | N | DOUBLE |  |  | 0 | Taxable income | 1000.00 |
| NonTaxable | Y | N | N | DOUBLE |  |  | 0 | Non-table payments (expenses etc) | 509.99 |
| IncomeTax | Y | N | N | DOUBLE |  |  | 0 | Income tax deducted | 123.99 |
| NationalInsurance | Y | N | N | DOUBLE |  |  | 0 | National Insurance deducted | 99.50 |
| NetPay | Y | N | N | DOUBLE |  |  | 0 | Net pay (payment made) | 1230.22 |
| Payday | Y | N | N | DATE |  |  |  | Payday for tax period | 2016-02-01 |
| TransferDay | Y | N | N | DATE |  |  |  | Date payment actually made | 2016-01-28 |
| TransferRef | Y | N | N | VARCHAR | 32 |  |  | Payment reference | TRX-99712-XBR |

**Key:** R = Required (Y/N), I = Indexed (Y/N), U = Unique (Y/N)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Database:** | Wombat HR | **Table Name:** | Contract | **Key Field(s):** | ContractID |

**Related to:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name** | **Foreign Key** | **Table Name** | **Foreign Key** |
| Employee | EmployeeID |  |  |

**General description:**

|  |
| --- |
| Employment contracts that employees have |

**Fields:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **R** | **I** | **U** | **Data Type** | **Length** | **Format Rule(s)** | **Default Value** | **Description** | **Example Data** |
| ContractID | Y | Y | Y | BIGINT |  |  |  | AUTO\_INCREMENT Contract ID primary key | 555 |
| Title | Y | Y | N | VARCHAR | 255 |  |  | Title of employment contract | “Consultant Permanent Contract” |
| Start | Y | N | N | DATE |  |  |  | Date begun | 2016-01-01 |
| DueFinish | N | N | N | DATE |  |  |  | Date due to finish | 2017-01-01 |
| ActualFinish | N | N | N | DATE |  |  |  | Actual date finished | 2016-06-13 |
| EmployeeID | Y | Y | N | BIGINT |  |  |  | FK EmployeeID on Employee.EmployeeID | 456 |

**Key:** R = Required (Y/N), I = Indexed (Y/N), U = Unique (Y/N)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Database:** | Wombat HR | **Table Name:** | EquipmentLoan | **Key Field(s):** | EmployeeID  EquipmentID |

**Related to:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name** | **Foreign Key** | **Table Name** | **Foreign Key** |
| Employee | EmployeeID | Equipment | EquipmentID |

**General description:**

|  |
| --- |
| Link table showing what equipment is or has been loaned to an employee |

**Fields:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **R** | **I** | **U** | **Data Type** | **Length** | **Format Rule(s)** | **Default Value** | **Description** | **Example Data** |
| EmployeeID | Y | Y | N | BIGINT |  |  |  | FK EmployeeID on Employee.EmployeeID | 456 |
| EquipmentID | Y | Y | N | BIGINT |  |  |  | FK EquipmentID on Equipment.EqupimentID | 789 |
| StartDate | Y | Y | N | DATE |  |  |  | Date loan started | 2016-02-01 |
| EndDate | N | Y | N | DATE |  |  |  | Date loan ended (equipment returned) | 2016-02-12 |
| Current | Y | Y | N | BOOLEAN |  |  | 1 | Current loan flag | 1 |
| Notes | N | N | N | TEXT |  |  |  | Notes regarding loan or return | “Returned heavily damaged” |

**Key:** R = Required (Y/N), I = Indexed (Y/N), U = Unique (Y/N)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Database:** | WombatHR | **Table Name:** | Equipment | **Key Field(s):** | EquipmentID |

**Related to:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name** | **Foreign Key** | **Table Name** | **Foreign Key** |
|  |  |  |  |

**General description:**

|  |
| --- |
| Equipment held within the company available for loan to employees |

**Fields:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **R** | **I** | **U** | **Data Type** | **Length** | **Format Rule(s)** | **Default Value** | **Description** | **Example Data** |
| EquipmentID | Y | Y | Y | BIGINT |  |  |  | AUTO\_INCREMENT primary key |  |
| Type | Y | Y | N | VARCHAR | 255 |  |  | Type of equipment | “Video Camera” |
| Make | N | N | N | VARCHAR | 255 |  |  | Manufacturer | “Sony” |
| Model | N | Y | N | VARCHAR | 255 |  |  | Model | “DXG-61B” |
| Description | N | Y | N | VARCHAR | 255 |  |  | Description of item | “Sony DV-8 digital video camera” |
| OperationNotes | N | N | N | TEXT |  |  |  | Specific operation notes | “On switch is the red thing” |
| Damaged | Y | Y | N | BOOLEAN |  |  | 0 | Damaged flag (not available for loan) | 1 |

**Key:** R = Required (Y/N), I = Indexed (Y/N), U = Unique (Y/N)

## Sample Data

Sample data will be provided separately through QOL after the start of the project.

Initially while creating the schema the emphasis is on the developer (you!) to build tests and suitable sample data to check the structure and constraints are working.

## Exploiting the Database

You are now required to demonstrate the power of the relational database you have implemented by writing queries to return the following data.

1. The EquipmentID, Make, Model, and Description of items of equipment that are (one query matching all the conditions):
   * Not currently marked as damaged
   * Not currently on loan
   * Have the word ‘video’ in their Type or Description
2. The EmployeeID, FirstName, and LastName of all current Employees who have a contract due to finish in 2019
3. A list of Project titles along with a column called Expenses showing the total expenses claimed against that Project (for no expenses NULL is fine as a value)
4. A list of employees showing FirstName, LastName, Email, and Grade (textual title) where they have a skill containing the word ‘video’ (one entry per employee only even if employees have multiple video skills)
5. A list of project titles and number of employees assigned to each project (one query matching all the conditions) where:
   * More than 2 employees are assigned
   * The employees are current
   * The project is not internal
6. A list of employees current or otherwise (EmployeeID, FirstName, LastName) who are not assigned to any projects
7. A list of all skills available (where one or more employee has that skill) in an output of Skill Name, Number of Skilled Employees
8. A list of employee IDs with a second column “Equipment Loans” showing the number of pieces of equipment currently on loan to them (note only employees with loans should be shown not employees with no loans)
9. As 8 but the second column is “All Time Loans” and is the total of all loans (current and historic) for that employee
10. Identifying for a given period payments made to employees who did not have a current contract (period for query is 2017 i.e. 01-01-2017 to 31-12-2017 inclusive) and outputting a list of EmployeeIDs and PayslipIDs for any found – note that for the date range the Due date for a contract to finish is fine to use

# B. Java Program Update

Some years back “Dodgy Dave’s Software Emporium” were contracted to create a Customer Relationship Management (CRM) system for WWI. They created a MySQL and Java-based textual interface, which is available as an Eclipse project.

Instructions on installing and developing with the CRM system are contained in its folder.

## Task

You have been asked to update the CRM system to do the following:

1. When a search is performed against a company both the CompanyName and the PersonName should be checked for the search term (currently only the CompanyName is searched against).
2. When a list of contacts against a company is shown on the screen those that have the visible flag unset (set to 0) should **not** be displayed.
3. When a contact is logged the user should be able to set if it should be visible or not (defaulting to being visible).

## Deliverables for Part B (Java Program Update)

1. A copy of your newly updated CRM project – called WombatCRM.
2. A word document (docx format) inside that directory showing screenshots of your new functionality working, along with any explanations of issues or missing functionality.

# C. Future Planning

## Task

You are required to produce two brief reports (500 words max each) detailing two areas.

## Deliverables for Part C (Future Planning)

A word document (docx) called FuturePlanning.docx with **both** report sections contained within it, each report should be suitably titled.

## Reports

1. Considering the schema devised and implemented in Part A as a starting point, what changes or evolutions would you make from a design and/or technical perspective to improve it?
2. The HR system needs to be totally secure and operate at high speed at a number of international company locations, how would you suggest the infrastructure is setup to run the database system to meet security, availability, and performance requirements?

# Mark Allocation Scheme

|  |  |
| --- | --- |
| Part A – Schema Implementation | 70% |
| Schema Implementation | 30% |
| Data Insertion SQL | 20% |
| Exploitation Queries | 20% |
| Part B - Java Program Update | 10% |
| Updates Completed | 8% |
| Correct Code | 2% |
| Part C – Future Planning | 20% |
| Report 1 – Evolution | 10% |
| Report 2 – Deployment | 10% |
| Penalties  (Note this is not an exhaustive list and does not include standard penalties such as late submission or plagiarism under the regulations) | |
| Incorrect submission format – minor errors[[1]](#footnote-1) | -5% |
| Incorrect submission format – major errors[[2]](#footnote-2) | -10% |

# Final Submission

You should have three sets of deliverables from Part A, Part B, and Part C (each with specific file and naming requirements as detailed above).

You must put all of these into a single zipped folder called <student\_number>.zip and submit through QOL.

Within the single folder there must be three sub-folders named “Part A”, “Part B”, “Part C” containing each of the deliverables for each stage.

So for example the following folder structure may be used:

12345678/

Part A/

schema.sql

data.sql

query.sql

Part B/

Wombat CRM/

Report.docx

src/

…

Part C/

FuturePlanning.docx

1. Minor submission format errors would be a small mistake in the folder structure or naming convention (for example schemas.sql rather than schema.sql etc) [↑](#footnote-ref-1)
2. A major error in submission format would be many incorrectly named or structured files requiring manual restructuring for marking – note that elements which are submitted entirely incorrectly (for example a Word document instead of an SQL file, etc) will be regarded as not submitted and not marked [↑](#footnote-ref-2)